Mutual influence of firing rates of corticomotoneuronal (CM) cells for learning a precision grip task

Octave Boussaton, Laurent Bougrain, Thierry Viéville, Selim Eskiizmirliler

Mutual influence of firing rates of corticomotoneuronal cells for learning a precision grip task

O. Boussaton, L. Bougrain, T. Vieville and S. Eskiizmirliler

Introduction

As a part of a Brain-Machine Interface, we define a model for learning and forecasting muscular activity, given sparse cortical activity in the form of action potential signals (spike trains). Whereas very impressive results such as [1] exist where a reaching task is successively performed from the sole interpretation of cortical signals, we focus our efforts in formalizing how neural impulses can be transcribed into a flexion of the index finger. We have a collection of experiments in which a trained monkey (*macaca nemestrina*) performs a precision grip. Its neuronal activity is partially recorded as the monkey clasps two levers between its index finger and thumb. In these experiments, 33 corticomotoneuronal (CM) cells from the hand area of the motor cortex (area 4) were recorded with glass-insulated platinum-iridium micro-electrodes, refer to [2] for more details about retrieving and filtering the data in our particular experiments. The main objective of this work is to treat the data in a way that allows us to provide an effective input/output functional. The underlying model parameters being interpreted with respect to the physiological aspects, though the model itself is not a bio-physical one. The method used here is based on a system of first degree linear equations involving the firing rate of the recorded neurons, two sets of thresholds associated to them, and the variation of the *global* neuronal activity. The learning formula is validated over a training set and tested over an estimation set.

Model description

The muscular activity of the index finger is related to the displacement of the lever that reads as a numerical value $p(t)$ over the experiment. We consider this as a trajectory we try to learn. The prediction we make at time $t > 0$ is noted $\hat{p}(t)$ and $\hat{p}(0) = p(0)$.
For every experiment, the recorded brain activity comes in the form of a series of spike train signals $n(t) = (n_1(t), n_2(t), ..., n_N(t))$ where N is the number of neurons. $\forall i \in [1, N]$ and at every time step t during the experiment, $n_i(t) = 1$ if an action potential has been detected or 0 otherwise. First, we compute the **firing rate function** of every neuron i according to a time-window of length w_i: $\forall i \in [1, N],\ d_i(t) = \sum_{k=t-w_i}^{t-1} n_i(t)$.
We call $d(t) = (d_1(t), d_2(t), ..., d_N(t))$ the **neuronal state** (of the subject) at time t.
Secondly, each neuronal state is associated to the *average of the derivative* values of observed index displacements, and also to the averaged previous state over a short period of time, for that specific neuronal state.
Finally, we use a synchrony information between each possible pair of neurons that is propagated for a certain period of time via **synchrony trains** and their firing rate functions, computed in the exact same way as the firing rate functions of the *original* spike trains:
let $n_{i,j}$ be the synchrony train of spike trains n_i and n_j, then $n_{i,j}(t) = 1$ if both neurons i and j emitted an action potential at time t, 0 otherwise.
For an experiment where N neurons are recorded, then C_N^2 $(= \frac{N\times(N-1)}{2})$ synchrony trains are computed, for simplicity, they will be written $d'_i(t)$ here for every $i \in [1, C_N^2]$.

Neuronal states

For each neuron $i \in [1, N]$, firing rate functions have a maximum value max_i. The space of all possible neuronal states is $D = \prod_{i=1}^N [0, max_i]$ then for any experiment, at all time t, its neuronal state $d(t)$ is in D.
Prior to learning, for every neuronal state $x = (x_1, x_2, ..., x_n)$ present at least once in the learning set, we record the average derivative value of the trajectory.
If $\#x$ is the number of times the neuronal state x appears in the learning set and $\dot{p}_1, \dot{p}_2, ..., \dot{p}_{\#x}$ all the observed values of the derivative of the trajectories in the learning set, then:

$$adv(x) = \frac{1}{\#x}\sum_{i=1}^{\#x} \dot{p}_i$$

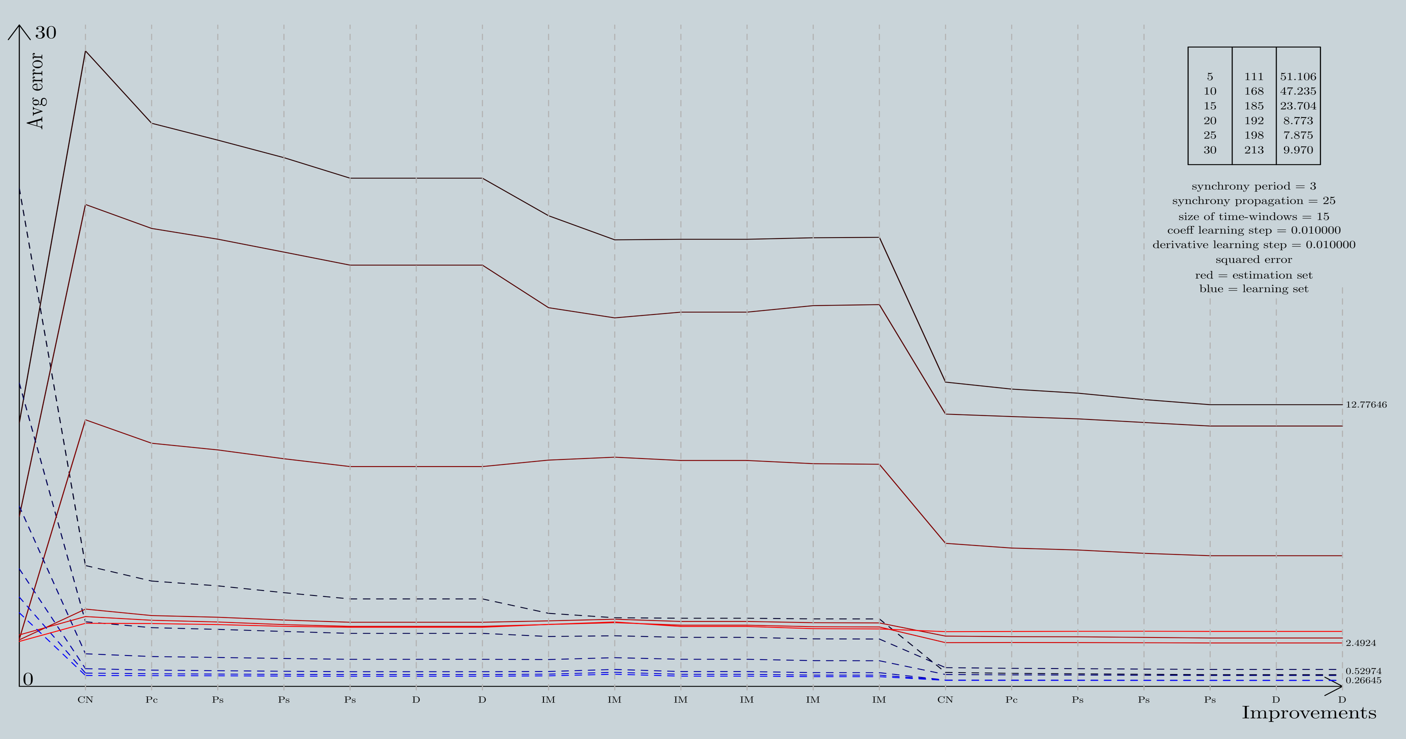
Mutual influence between neurons

$$\begin{aligned}\epsilon(d(t)) = &\sum_{i=1}^N \beta_i \frac{d_i(t)}{\delta_i} \times \Big(\sum_{j=1}^N \gamma_{i,j} \frac{d_i(t)}{\phi_{i,j}} + \sum_{j=1}^{C_N^2} \delta_{i,j} \frac{d'_i(t)}{\phi'_{i,j}}\Big) + \\ &\sum_{i=1}^{C_N^2} \beta'_i \frac{d'_i(t)}{\delta'_i} \times \Big(\sum_{j=1}^N \gamma'_{i,j} \frac{d_i(t)}{\phi_{i,j}} + \sum_{j=1}^{C_N^2} \delta'_{i,j} \frac{d'_i(t)}{\phi'_{i,j}}\Big) + dist(ans(d(t)), d(t)).\end{aligned}$$

where $\sum_{i=1}^N \gamma_{i,j} + \sum_{i=1}^{C_N^2} \delta_{i,j} = \sum_{i=1}^N \gamma'_{i,j} + \sum_{i=1}^{C_N^2} \delta'_{i,j} = 1$. δ_*, ϕ_* and ϕ'_* are thresholds, all of which are variables that undergo adjustment while learning and $dist(a, b)$ is the Euclidian disance between a and b.

Learning efficiency

Improvements can be done independently on each type of parameters. On this picture, blue curves are those of the learning set, in red are those of the estimation set.



Note the over-learning here, best results are obtained with 25 trajectories in the learning set.

Output prediction

$\forall t > 0$, the prediction $\hat{p}(t+1) = \hat{p}(t) + adv(d(t)) \times A(d(t)) + \epsilon(d(t))$

where $adv(t)$ is the average value of the derivative for neuronal state $d(t)$ if this state exists in the learning set, 1 otherwise, $A(d(t))$ is a ponderation functional and $\epsilon(t)$ is an adjustment based on linear combinations of the firing rates of the neurons.

Pondering the variation of the trajectory

$$A(d(t)) = \sum_{i=1}^N \alpha_i \frac{d_i(t)}{\theta_i} + \sum_{i=1}^{C_N^2} \alpha'_i \frac{d'_i(t)}{\theta'_i} \quad \text{with}$$

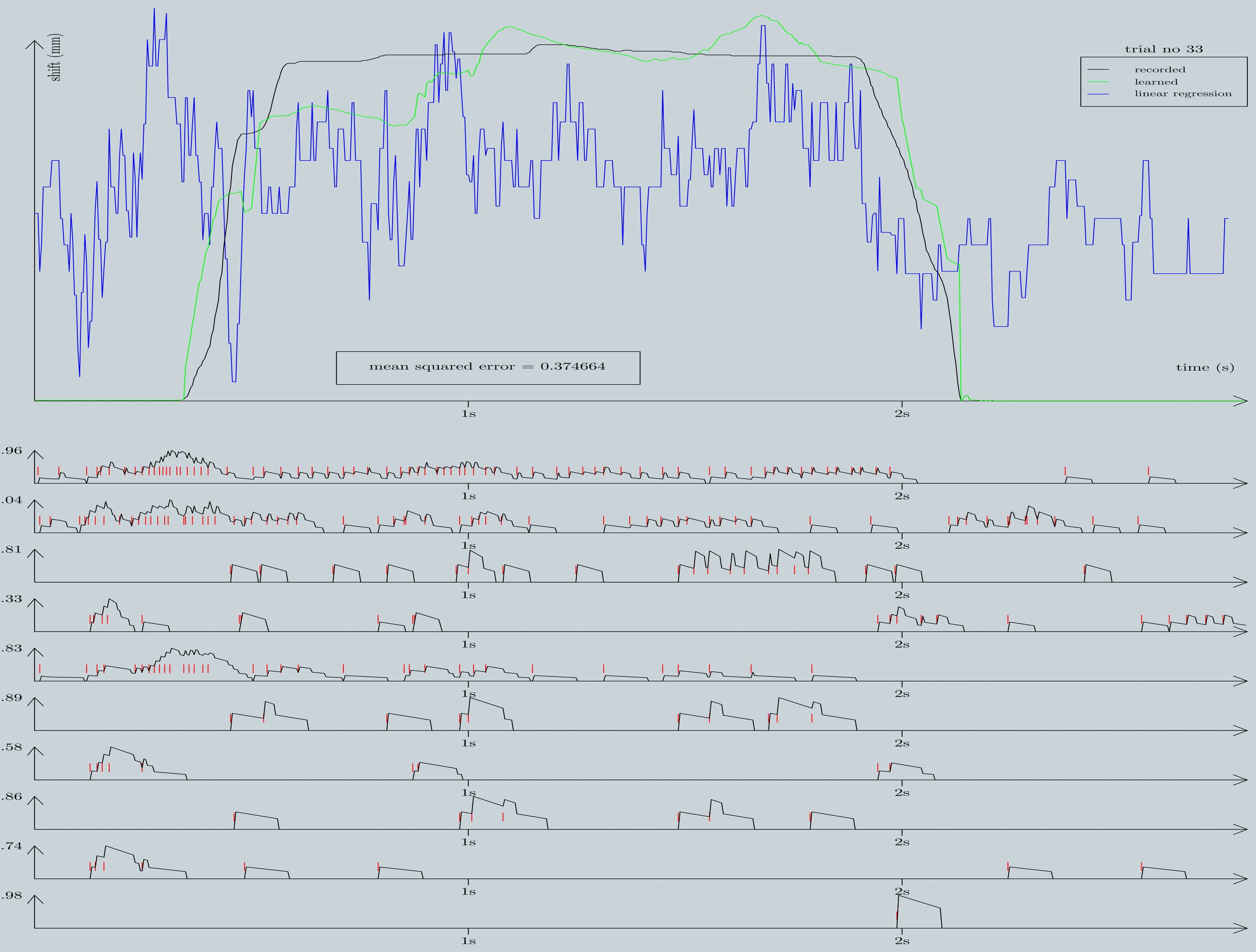
$\sum_{i=1}^N \alpha_i = 1, \sum_{i=1}^{C_N^2} \alpha'_i = 1$. θ_* and θ'_* are threshold values optimized during learning.

Averaged previous neuronal state

Similarly to the averaged derivative value, the averaged previous neuronal states are also defined according to a time window, let w_s be its length.
We write $m_i^s(d(t)) = \frac{1}{s}\sum_{k=t-s}^{t-1} d_i(k)$ the previous average state of period s for neuron i at time t.
For every neuronal state $x = (x_1, x_2, ..., x_n)$ present at least once in the learning set, we record its averaged previous neuronal state.
If $\#x$ is the number of times the neuronal state x appears in the learning set and $x_1, x_2, ..., x_{\#x}$ all its occurences in the learning set, then:

$$ans_i(x) = \Big(\frac{1}{\#x}\sum_{i=1}^{\#x} m_1^s(x_i), ..., \frac{1}{\#x}\sum_{i=1}^{\#x} m_N^s(x_i)\Big)$$

Results



The result here has been obtained with 30 experiments in the learning set and 9 for evaluation. The example above is the most innacurate result (green curve). We used a linear decay in the calculation of the spiking functions and a window size of 40 milliseconds. The four upper spike trains are of recorded brain activity, the ones below are synchrony calculations.
This shows the limits of the method but we also had a small number of recorded neurons available per experiments. The blue curve is a learning done through linear regression for comparison.

References & acknowledgements

[1] M. Velliste, S. Perel, S. Chance, A. Spalding, A. Schwartz Cortical control of a prosthetic arm for self feeding In *Nature 2008: 453: 1098-1101*
[2] M. Maier, K. Bennett, M. Hepp-Raymond, R. Lemon Contribution of the monkey corticomotoneuronal system to the control of force in precision grip In *Journal of neurophysiology 1993: 18(3): 772-785*

This work was in part supported by CNRS-PIR-Neuro-IC 2010 (Project:CorticoRobot). We are also very thankful to R. Lemon and M. Maier for providing the data set to us.